

Министерство просвещения Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Уральский государственный педагогический университет»
Институт математики, физики, информатики и технологий
Кафедра информатики, информационных технологий
и методики обучения информатике

МОБИЛЬНАЯ СИСТЕМА ПИРИНГОВОГО ОЦЕНИВАНИЯ

*Выпускная квалификационная работа
бакалавра по направлению подготовки
09.03.02 – Информационные системы и технологии*

Исполнитель: студент группы ИСиТ-1601
Института математики, физики,
информатики и технологий
Аверин А.В.

Допустить к защите
«_____» _____ 2020 г.

Зав. кафедрой _____
М.В. Лапенюк

Руководитель ОПОП _____
Л.В. Сардак

Руководитель: д.п.н., профессор
кафедры ИИТиМОИ
Стариченко Б.Е.

Екатеринбург – 2020

РЕФЕРАТ

Аверин А.В. МОБИЛЬНАЯ СИСТЕМА ПИРИНГОВОЙ ОЦЕНКИ,
выпускная квалификационная работа: 50 стр., рис. 22, библи. 35 назв.

Ключевые слова: ПИРИНГОВАЯ ОЦЕНКА, ВЕБ-ПРИЛОЖЕНИЕ, МОБИЛЬНАЯ СИСТЕМА, МЕТОДЫ ОЦЕНКИ, СРЕДСТВА ОЦЕНКИ, ОБРАЗОВАНИЕ.

Предмет разработки – веб-приложения для осуществления мобильной пиринговой оценки.

Цель работы – разработать кроссплатформенное веб-приложение реализующее метод пирингового оценивания.

В настоящей работе описаны результаты проектирования и программной реализации веб-приложения для пирингового оценивания, позволяющего проводить мобильную пиринговую оценку в любых местах, где есть доступ в сеть интернет и обучающиеся могут подключиться со своих мобильных устройств.

Система была реализована посредством клиент-серверной архитектуры, с использованием языка программирования JavaScript, как на серверной части, так и на клиенткой. В качестве веб-фреймворка для серверной части использовался Express.js, клиентская часть и интерфейс приложения разработаны на основе библиотеки React, в качестве базы данных выбрана документоориентированная система управления базами данных MongoDB.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
ГЛАВА 1. РОЛЬ ПИРИНГОВОГО ОЦЕНИВАНИЯ В ОБРАЗОВАНИИ 5	
1.1 ТЕХНОЛОГИИ ДИСТАНЦИОННОГО ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ.....	5
1.2 АНАЛИЗ ТЕХНОЛОГИЙ РЕАЛИЗАЦИИ ДИСТАНЦИОННОГО ОЦЕНИВАНИЯ	14
1.3 ТЕХНИЧЕСКОЕ ЗАДАНИЕ.....	20
ГЛАВА 2. РЕАЛИЗАЦИЯ МОБИЛЬНОЙ СИСТЕМЫ ПИРИНГОВОГО	
ОЦЕНИВАНИЯ	31
2.1 ОПИСАНИЕ АЛГОРИТМОВ И ПРОГРАММНОЙ РЕАЛИЗАЦИИ	31
2.2 ОПИСАНИЕ ФУНКЦИОНАЛЬНЫХ ВОЗМОЖНОСТЕЙ И ИНТЕРФЕЙСА ПРИЛОЖЕНИЯ	
ERROR! BOOKMARK NOT DEFINED.	
2.3 ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ РАЗРАБОТАННОЙ СИСТЕМЫ ERROR! BOOKMARK NOT	
DEFINED.	
ЗАКЛЮЧЕНИЕ	44
СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ.....	45
ПРИЛОЖЕНИЯ	49
<i>ПРИЛОЖЕНИЕ 1.....</i>	49
<i>ПРИЛОЖЕНИЕ 2.....</i>	50

ВВЕДЕНИЕ

С развитием информационно-коммуникационных технологий (ИКТ) в сфере образования начали появляться новые и эффективные инструменты для работы с обучающимися. Многочисленные программы, сервисы и сайты реализуют различные методы оценки посредством ИКТ. Однако, не смотря на все многообразие и функциональность, остаются методы, реализация которых (посредством ИКТ) может оказаться если не невыполнимой, то крайне трудной. Один из таких: метод пирингового оценивания.

В связи с этим представляется актуальным разработка веб-приложения, которое позволит производить мобильную пиринговую оценку в любом месте, где обучающиеся смогут подключиться к сети интернет со своих мобильных устройств.

Предмет разработки: веб-приложение для мобильной пиринговой оценки.\

Цель: спроектировать и разработать веб-приложение позволяющее проводить мобильную пиринговую оценку.

Задачи:

1. Провести анализ информационных источников для выявления существующих подходов к реализации метода пиринговой оценки посредством информационно-коммуникационных технологий.
2. Проанализировать возможности существующих программных продуктов для реализации метода пиринговой оценки, на основе чего обосновать необходимость разработки своего продукта. Также обосновать выбор технологий реализации и необходимых программных платформ.
3. В соответствии с техническим заданием провести разработку веб-приложения для пиринговой оценки.
4. Подготовить техническую и сопроводительную документацию.

ГЛАВА 1. Роль пирингового оценивания в образовании

1.1 Технологии дистанционного оценивания результатов обучения

Средства оценки результатов обучения – это технологии, предоставляющие обработку и хранение информации об успеваемости обучающихся. Для преподавателя такие технологии дают возможность наблюдать за тем, как продвигается учебный процесс, а также рассмотреть и произвести анализ работы каждого в отдельности взятого обучающегося. Для учащихся происходит формирование адекватной оценки собственной успеваемости, что влияет на повышение мотивации к обучению, однако это происходит, если поставленная отметка объективна. Чтобы отметка была объективной, необходимо использовать современные методы оценки результатов обучения.

Оценка – это практически единственный, доступный преподавателю инструмент мотивации, и чем она объективней, тем серьезнее отношение обучающихся к своим успехам. В связи с этим, методы и средства оценки обучения необходимо постоянно совершенствовать.

Большой толчок в развитии средств и методов оценивания произошел с появлением информационных и коммуникационных технологий. Оценка посредством ИКТ оказалась настолько удобной и продуктивной, что ИКТ-компетентность включили в образовательные и профессиональные (педагогические) стандарты. [1, 19]

По приказу Министерства труда и социальной защиты Российской Федерации от 18 октября 2013г. №544н. Профессиональный стандарт. Педагог (педагогическая деятельность в дошкольном, начальном общем, основном общем, среднем общем образовании) (воспитатель, учитель).

Пункт 3.1.1. Трудовая функция. «Общепедагогическая функция. Обучение»

Трудовые действия:

- Формирование навыков, связанных с информационно-коммуникационными технологиями (далее – ИКТ)
- Объективная оценка знаний обучающихся на основе тестирования и других методов контроля в соответствии с реальными учебными возможностями детей

Необходимые умения:

- Объективно оценивать знания обучающихся на основе тестирования и других методов контроля в соответствии с реальными учебными возможностями детей
- Владеть ИКТ-компетентностями: общепользовательская ИКТ-компетентность; общепедагогическая ИКТ-компетентность; предметно-педагогическая ИКТ-компетентность (отражающая профессиональную ИКТ-компетентность соответствующей области человеческой деятельности)

Необходимые знания:

- Основы методики преподавания, основные принципы деятельностного подхода, виды и приемы современных педагогических технологий

Пункт 3.1.3. Трудовая Функция. «Развивающая деятельность»

Трудовые действия:

- Применение инструментария и методов диагностики и оценки показателей уровня и динамики развития ребенка

Пункт 3.2.1. Трудовая Функция. «Педагогическая деятельность по реализации программ дошкольного образования»

Необходимые умения:

- Владеть ИКТ-компетентностями, необходимыми и достаточными для планирования, реализации и оценки образовательной работы с детьми раннего и дошкольного возраста

Приказ Минобрнауки России от 22.02.2018 №126 "Об утверждении федерального государственного образовательного стандарта высшего образования - магистратура по направлению подготовки 44.04.01 Педагогическое образование.

1.5. При реализации программы магистратуры Организация вправе применять электронное обучение, дистанционные образовательные технологии.

1.6. Реализация программы магистратуры осуществляется Организацией как самостоятельно, так и посредством сетевой формы.

В данный момент существует огромное множество подходов к совершенствованию средств и методов дистанционного оценивания результатов обучения с применением ИКТ.

Рассмотрим некоторые:

1. Система управления обучением (LMS - Learning management system)

Под системой управления обучением понимают программный продукт (приложение, сервис, сайт) для администрирования учебных курсов в рамках дистанционного обучения. [23] LMS это очень мощный инструмент в руках преподавателя, и является связующим звеном для большинства средств оценивания результатов обучения. Из уже существующих LMS можно выделить: ЯКласс, Edmodo, Google Класс.

- ЯКласс – цифровой образовательный ресурс для школ. Сайт www.yaklass.ru появился в марте 2013 года и на сегодняшний день стал площадкой для более чем пятидесяти тысяч школ в России, Украине, Беларуси, Латвии, Армении и Австрии. ЯКласс позволяет составлять проверочные работы для учащихся с индивидуальными заданиями. Проверку и отчетность можно автоматизировать. Система позволяет отследить тенденции в обучении каждого отдельного ученика. Есть большое количество уже готовых заданий и материалов для школьников.

Из минусов можно выделить, что приложение платное и упор делается больше на школьников (Не самая подходящая LMS для студентов).

- Edmodo – соц.сеть для школьников и студентов. При работе с этой системой преподаватели и учащиеся создают бесплатные аккаунты. Преподаватель создает группу (класс). У одного преподавателя может быть несколько групп, и каждый учащийся может состоять в нескольких группах. Когда учащиеся входят в Edmodo, то видят ленту новостей, то есть все объявления и записи от групп, в которых они состоят. Преподаватели могут хранить в системе доступные для скачивания учащимися учебные материалы, собирать и оценивать задания, выдавать значки за успехи. Главным минусом системы является отсутствие русской локализации.
- Google класс позволяет создавать учебные классы, публиковать и оценивать задания, общаться с учащимися, размещать объявления, а также хранить и распространять учебные материалы. Также из-за интеграции с другими инструментами от Google (Google диск / документы / таблицы / презентации / календарь / почта) – Google класс становится очень мощным и гибким инструментом. Основным минус это ограниченные возможности для группового обсуждения.

2. Прокторинг

Система прокторинга предназначена для контроля над учащимися на предмет самостоятельного прохождения онлайн экзаменов или тестов в системах дистанционного обучения или в самой системе прокторинга, которая позволяет создавать тесты. Система верифицирует личность участника и наблюдает за ним в режиме реального времени для выявления возможных нарушений. Для использования прокторинга студенту потребуется: подключение к интернету, компьютер с микрофоном и камерой, также может потребоваться мобильный телефон с камерой для наблюдения за обучающимся

с нескольких ракурсов. Организация, в свою очередь, должна иметь продукт, в котором будет проводиться экзамен с прокторингом.

Прокторинг позволяет провести дистанционное тестирование обучающихся, в удобном формате, и получить в результате валидные оценки. Минусом является сложность реализации: не у всех студентов может быть необходимое оборудование. Также продукты прокторинга и их поддержание могут оказаться дорогостоящими (в зависимости от количества тестируемых в один момент времени учащихся).

3. Мобильный опрос и тестирование

Тестирование - это один из главных методов оценки, а с появлением ИКТ его реализация в учебном процессе стала максимально простой, удобной и эффективной. Преподавателю необходимо всего лишь составить тест или опрос в удобном конструкторе и сообщить ученикам его адрес, на который необходимо перейти при помощи мобильного телефона, компьютера или ноутбука. Преподавателю доступна вся статистика по каждому ученику и классу в целом. Также в случае с тестированием приложение самостоятельно проверит, на какое количество вопросов ученики дали верный ответ, и составит список с высчитанными отметками. Существует большое количество программ и веб-приложений для мобильных опросов и тестирования. Можно выделить несколько веб-приложений: Socrative и Google Forms.

- Socrative - это бесплатное приложение для создания и проведения опросов или тестов. С помощью него можно создать тесты с различными типами ответов: выбор одного или нескольких ответов из предложенных, написать ответ самостоятельно, выбрать правда или ложь. Также в Socrative можно добавить игровой момент - запустить «Гонку». Можно создать до 20 команд и победит та, которая даст больше правильных ответов и первой дойдет до конца.

Во вкладке results сформируется статистика со всеми переданными ответами.

☒ Show Names

☒ Show Answers

Name ↑	Score (%) ▼	1	2	3
AAAAA	33%	C, A	True	cat
Andrew	67%	A	False	cat
Class Total		0%	50%	100%

Click question numbers or class total percentages for detailed views.

Рис. 1. Приложение Socrative. Таблица результатов

Приложение Socrative – это мощный инструмент со всем необходимым функционалом в бесплатной версии. Единственный минус, это отсутствие русской локализации интерфейса.

- Google Формы – это онлайн инструмент для создания тестов и форм обратной связи. Инструмент от Google хорошо визуализирует результаты: строит круговые диаграммы и гистограммы по полученным ответам. Имеет большой выбор форматов для ответов: текст; один из списка; несколько из списка; раскрывающийся список; загрузка файлов; шкала; дата; время и так далее. Google Формы Обладают большими возможностями за счет интеграции с другими инструментами от Google. По одному нажатию экспортирует ответы в excel (Google таблицы). Также форму с тестом можно добавить в LMS от Google (Google класс) как задание и все результаты запишутся в таблицу-журнал.

Для подавляющего числа методов оценки существуют средства их реализации посредством ИКТ. Компании, которые разрабатывают данные продукты, стараются сделать их многофункциональными (один продукт может быть использован для реализации нескольких методов и множества других

задач) и зачастую основным предназначением продукта является даже не реализация какого-либо метода оценки.

Всё-таки остаются методы, реализация которых (посредством ИКТ) может оказаться если не невыполнимой, то крайне трудной. Одним из таких методов является пиринговое оценивание.

Пиринговое оценивание - это способ проверки и оценивания работ, когда работа одного автора оценивается несколькими независимыми экспертами по заданным критериям. Роль экспертов на себя берут одноклассники, однокурсники или коллеги (люди из одного с автором круга).

Пиринговое оценивание помогает студенту научиться делать умозаключения и оценивать результат своего труда и труда других людей в некоторой предметной области.

В педагогике хорошо известна и применяется на практике классификация педагогических целей обучения Б. Блума (таксономия Блума). Цели имеют иерархическую структуру (рис. 2), и процесс освоения учебной информации можно рассматривать как продвижение вверх от самой простой цели – получить знания – к более высоким. [25]



Рис. 2. Таксономия целей обучения Б. Блума («пирамида» Блума)

Можно заметить, что наивысшим уровнем в предложенной иерархии является «Оценивание». Соответственно в процессе оценивания студент должен обладать необходимым профессиональным уровнем подготовки.

В настоящее время взаимное оценивание («пиринговое оценивание») обучающимися работ своих сокурсников получило развитие в связи с широким распространением массовых открытых онлайн курсов (МООК), в которых данная форма используется наряду с традиционным тестовым контролем, проектами и заключительным тестированием с применением систем прокторинга. Однако и в МООК используется только письменный вариант оценивания[15]. В то же время, можно представить множество учебных ситуаций, в которых было бы желательно осуществлять взаимное оценивание в режиме реального времени: при публичной защите проектов (в том числе, дистанционной), выступлении на семинаре, представлении своих учебных продуктов (программ, графики и видео, творческих работ). При этом оценивание не должно занимать много времени и, следовательно, не быть письменным, и должно быть осуществимо в любой (неспециализированной) аудитории или даже вне ее.

Как было сказано ранее: Пиринговая оценка труднореализуема с помощью существующих программных продуктов, но постараться ее реализовать всё-таки можно с помощью Google форм и Google таблиц.

Google формы позволяют нам добавлять неограниченное количество вопросов. В терминологии пирингового оценивания «вопросы» будут называться «критериями», а «форма» - «протокол оценки». После создания протокола мы отправляем ссылку на него всем обучающимся. И собираем оценки по всем авторам от экспертов.

Рис. 3. Реализация пиринговой оценки в Google формах

Созданный протокол выглядит желанным образом, но полученные результаты формируются не так, как хотелось бы. Вместо оценок от всех экспертов по каждому отдельному критерию, мы получаем транспонированный вариант - оценки по всем критериям от каждого эксперта.

	A	B	C	D	E	F	G	H
1	Отметка времени	Адрес электронной почты	Баллы	Ваше имя:	Имя автора работы	Первый критерий оценки	Второй критерий оценки	Третий критерий оценки
2	04.04.2020 19:34:44	ex1@uspu.su	0 / 4	эксперт 1	автор 1	1	3	2
3	04.04.2020 19:35:59	ex2@uspu.su	0 / 4	эксперт 2	автор 1	3	3	1
4	04.04.2020 19:36:02	ex3@uspu.su	0 / 4	эксперт 3	автор 1	3	3	1
5	04.04.2020 19:36:20	ex4@uspu.su	0 / 4	эксперт 4	автор 1	2	2	3
6	04.04.2020 19:36:39	ex5@uspu.su	0 / 4	эксперт 5	автор 1	2	3	1
7	04.04.2020 19:36:55	ex6@uspu.su	0 / 4	эксперт 6	автор 1	2	2	2
8	04.04.2020 19:37:13	ex7@uspu.su	0 / 4	эксперт 7	автор 1	3	2	1
9								

Рис. 4. Полученная таблица результатов из Google форм экспортированная в Google таблицы.

Полученную таблицу необходимо отфильтровать по авторам и проделать множество ручной работы: транспонировать таблицу, прописать формулы

высчитывания среднего балла по каждому критерию и высчитывания общей оценки.

Также можно вручную создать протоколы для каждого автора в Google таблицах и как ранее прописать все формулы. Однако есть существенные минусы:

- эксперты смогут видеть, какие оценки поставили их коллеги;
- эксперт может случайно или намерено изменить оценки других авторов (можно защитить таблицу от редактирования и выдать права каждому эксперту на выделенный под его оценку диапазон, но для этого необходимо знать имеющуюся Gmail почту эксперта)

Автор 1					
Ваше имя:	эксперт 1	эксперт 2	эксперт 3	эксперт 4	эксперт 5
Первый критерий оценки	3	3	3	2	3
Второй критерий оценки	2	1	1	3	1
Среднее	5	4	4	5	4
Итоговый бал:					4,4

Рис. 5. Протокол оценки созданный посредством Google таблиц.

Рассмотрев одни из возможных вариантов реализации можно сделать вывод: имеющиеся программные продукты, посредством которых можно реализовать пиринговую оценку: плохо решают поставленную задачу. Отсутствие автоматизации делает их трудозатратными и повышает порог вхождения (требует от пользователя определенных знаний и навыков).

В связи со сказанным представляется актуальной разработка веб-приложения, которое обеспечило бы возможность проведения пирингового оценивания многими экспертами с мобильных устройств; при этом результаты оценивания должны быть доступны преподавателю с некоторого сетевого хранилища, но нужно выбрать оптимальные подходы к реализации.

1.2 Анализ технологий реализации дистанционного оценивания

Продукт, реализующий средство пирингового оценивания, выполняет следующие, основные, задачи:

- Создание протоколов оценки;
- Хранение созданных протоколов и полученных по ним оценок;
- Возможность подключения к протоколу оценки не менее 30 экспертов;
- Формирование таблицы результатов по полученным оценкам;
- Доступ с различных современных устройств: мобильных телефонов, планшетов, ноутбуков, стационарных компьютеров.

Таким образом, приложение получает, хранит, обрабатывает и возвращает полученные от пользователя данные. Все это привело к тому, что была использована клиент-серверная архитектура, а необходимость подключения не только со стационарных, но и мобильных устройств подтолкнуло к разработке web-приложения, т.к. практически на любом современном устройстве есть браузер и доступ в интернет.

Клиент-серверная архитектура – это сетевая архитектура, в которой и клиент и сервер представляют собой программное обеспечение, расположенное на разных вычислительных устройствах и общаются эти устройства посредством сетевых протоколов (HTTP, FTP и др.). [14] Клиент отправляет запрос на сервер; Сервер в свою очередь обрабатывает запрос, обращается при необходимости к базе данных, и возвращает ответ.



Рис. 6. Схема клиент-серверной архитектуры

При разработке web-приложения клиентом будет являться браузер на вычислительном устройстве пользователя, а сервер - вычислительная машина способная получить HTTP запрос, обработать его и вернуть ответ клиенту.

Практически на любом языке программирования можно реализовать серверную часть приложения и связать с СУБД, однако, некоторые языки

являются более подходящими, под них написаны библиотеки и веб-фреймворки для работы с HTTP протоколом. Таким образом первой задачей был выбор языка программирования для серверной части. Самыми востребованными и признанными языками в области разработки веб-сервисов и веб-приложений на данный момент можно назвать: Python, PHP, JavaScript (Node.js).

Если рассматривать Python, то это отличный язык на котором можно писать практически все. Для разработки веб-приложений и веб-сервисов на Python существует фреймворк Django. Django может использоваться с любым фронтенд фреймворком. Он также представляет собой хорошее решение для любых типов веб-приложений из-за множества доступных сторонних пакетов.

PHP – скриптовый язык общего назначения. PHP был создан для разработки веб-приложений и в данный момент подавляющее большинство хостинг-провайдеров поддерживают его. Также множество веб-сервисов и веб-приложений написаны именно на нем, в связи с этим в интернете есть огромное количество информации и готовых решений. Однако из-за странного синтаксиса и отсутствия строгой типизации – получается некачественный код, который, с ростом проекта, трудно поддерживать.

Node.js – программная платформа, основанная на движке V8. Node транслирует JavaScript в машинный код, тем самым делая его языком общего назначения (дает возможность при помощи JavaScript взаимодействовать с устройствами ввода-вывода и подключать написанные на других языках библиотеки). С Node легче масштабироваться. Если к серверу подключатся тысячи пользователей, то асинхронный Node распределит нагрузку, а не выделит на каждую задачу отдельный поток. Также в состав Node.js входит менеджер пакетов Npm, в котором имеется огромное количество внешних библиотек и готовых модулей. Ко всему этому за последние несколько лет JavaScript очень сильно вырос. В нем появились: классы, модули и загрузчики модулей, хорошая работа с асинхронностью посредством таких инструментов, как Promise и Async/Await и многое другое.

Для написания серверной части приложения отлично подошел бы любой из рассмотренных языков. Однако для написания клиентской части необходимо использовать JavaScript, т.к. это сильная сторона языка и до появления Node всегда была основной его задачей. В связи с этим разрабатывать серверную часть и связывать ее с клиентской, будет проще используя Node.js и веб-фреймворк Express. Также пакетный менеджер Npm сильный инструмент упрощающий разработку.

Для написания клиентской части, как уже было сказано, был использован JavaScript, но ее написание на чистом JS – трудозатратно и не всегда эффективно. Нужно использовать JavaScript библиотеки или фреймворки для разработки пользовательских интерфейсов. React, Angular и Vue.js на данный момент лидирующие продукты для разработки UI.

React – технология разработанная компанией Facebook, что безусловно является большим плюсом, потому как поддержка такой большой компанией говорит о том, что продукт будет развиваться и оставаться актуальным еще долгое время. Также React имеет отличную документацию и огромное сообщество пользователей.

Vue.js – простой фреймворк, за которым стоит всего один разработчик Эван Ю. Vue был создан с расчетом на то, что не существует готовых решений для быстрого прототипирования сложных пользовательских интерфейсов веб-приложений, т.к на тот момент существовал только сложные фреймворки ориентированный на разработку больших приложений. В итоге получился легковесный и простой в понимании фреймворк.

Angular – большой фреймворк от Google основанный на TypeScript для разработки больших приложений. Один из главных плюсов Angular его насыщенность: он изначально включает в себя практически все необходимые пакеты и модули. С другой стороны это может быть и минусом, так как в связи с этим Angular становится тяжеловесным.

В связи с тем, что разрабатываемое веб-приложение будет относительно небольшим, тяжеловесность Angular могла бы привести к проблемам скорости загрузки пользователей – это сделало его не подходящим для разработки. Выбирая между React и Vue.js можно было бы остановиться на любом из них, но React на данный момент, имеет чуть больше функциональных возможностей и отличается качеством документации.

Таким образом, остается лишь обосновать выбор базы данных. Для начала необходимо было определиться с моделью: реляционная или нереляционная.

При разработке базы данных потребовались следующие сущности для таблиц/коллекций: пользователь, протоколы, критерии оценки, полученные оценки. Данные сущности достаточно просто связываются между собой посредством реляционной модели: пользователь имеет множество протоколов (один ко многим), каждый протокол в свою очередь имеет некоторое количество критериев оценки (один ко многим), а протокол имеет множество оценок (один ко многим).

В нереляционной модели связь будет выглядеть следующим образом: коллекция пользователей имеет в себе ссылку на коллекцию протоколов, протоколы имеют в себе ссылку на коллекцию оценок. Сущность «критерии оценки» проще определить, как массив объектов внутри коллекции протоколов.

При использовании нереляционной модели часто возникают проблемы с нагрузкой на базу, т.к. запрашивая одну из коллекций может вернуться огромный массив данных. В нашем же случае данная проблема не актуальна, так как маловероятно, что пользователь будет создавать тысячи протоколов или будет получать на них тысячи оценок. Таким образом использование нереляционной модели оправданно.

Среди небольшого количества нереляционных баз данных большей популярностью обладает MongoDB, также для нее написана библиотека Mongoose. Mongoose – это объектно-документный отображитель, который

предоставляет огромный набор функциональных возможностей для создания и работы со схемами MongoDB.

Исходя из всего выше сказанного, был сформирован следующий стек технологий для реализации веб-приложения пиринговой оценки:

- язык программирования JavaScript;
- JavaScript-библиотека с открытым исходным кодом для разработки пользовательских интерфейсов React;
- программная платформа Node.js;
- фреймворк веб-приложений Express;
- база данных MongoDB и библиотека Mongoose для работы с ней.

1.3 Техническое задание

на разработку мобильной системы «Peer Assessment online»

Ниже представлено техническое задание на разработку мобильной системы, пирингового оценивания основанное на следующих документах:

- ГОСТ 34.602-89 [1];
- ГОСТ 19.201-78 [2].

1. Общие сведения.

1.1. Название организации-заказчика.

ФГБОУ ВО «УрГПУ»

1.2. Название продукта разработки.

Рабочее название продукта – «Peer Assessment online».

1.3. Назначение продукта.

Обеспечение возможности пирингового оценивания в любой аудитории с помощью мобильных устройств с формированием протокола оценивания по заранее определенным (преподавателем) критериям и его сохранением в облачном пространстве с возможностью его последующего предъявления аудитории или печати.

1.4. Термины и определения.

Эксперт – пользователь приложения, выполняющий оценку работ.

Автор – обучающийся, чью работу оценивают эксперты.

Протокол – список содержащий сгруппированные критерии оценки

Вес критерия – уровень значимости критерия относительно других критериев. Вес критерия используется при вычитывании среднего балла за работу автора.

Пиринговая оценка – способ проверки и оценивания работ, когда работа одного автора оценивается несколькими независимыми рецензентами (экспертами), обычно из одного с автором круга.

1.5. Плановые сроки начала и окончания работ.

Этап	Содержание работ	Сроки	Ответственный
1. Составление технического задания	Разработка функциональных и нефункциональных требований к системе	-	Исполнитель
2. Техническое проектирование	Разработка сценариев работы системы	В течение 28 дней после дня утверждения технического задания	Исполнитель
	Разработка макетов пользовательского интерфейса		
3. Разработка программной части	Разработка серверной части приложения	В течение 56 дней после дня согласования технического проекта	Исполнитель
	Разработка клиентской части приложения		
	Описание базы данных		
4. Ввод в эксплуатацию		-	Исполнитель

2. Характеристика области применения продукта

2.1. Процессы и структуры, в которых предполагается использование продукта разработки.

Основные структуры, в которых предполагается использование продукта, - учебные заведения (школы, институты). Также продукт может быть полезен для любых других структур нуждающихся в проведении экспертной оценки.

2.2. Характеристика персонала.

Пользователь с умением работать в браузере на любом устройстве.

3. Требования к продукту разработки

3.1. Требования к системе в целом.

Продукт должен представлять собой веб-приложение реализованный посредством клиент-серверной архитектуры.

3.2. Аппаратные требования.

Для ПК:

- Операционная системой Windows XP и более поздние версии; MacOS X 10.5 и более поздние версии; Ubuntu 14.04 (64-разрядная версия) или старше; Debian 8 или старше; openSUSE 13.3 или старше; Fedora Linux 24 или старше.

- Процессор Intel Pentium 4 или более поздней версии с поддержкой SSE2.
- От 64 МБ оперативной памяти.
- От 50 МБ свободного места на жестком диске.

Для мобильного телефона:

- Операционная система Android версии 4.4 и старше; iOS версии 6.1.6 и старше.

3.3. Необходимое системное программное обеспечение.

Для ПК:

- Один из браузеров: Edge, Safari, Opera, Google, Яндекс.Браузер, Chrome или Mozilla Firefox последних версий.

для мобильного телефона:

- Один из браузеров: Chrome for Android, Android Browser или iOS Safari последних версий.

3.4. Программное обеспечение, используемое для реализации.

- Visual Studio Code – интегрированная среда разработки;
- Браузер Google Chrome версии 81.0.4044.138;
- Postman – HTTP-клиент для тестирования API.

3.5. Особенности реализации серверной и клиентской частей.

Сервер должен принимать запросы по протоколу HTTP, обращаться при необходимости к базе данных и возвращать ответ на клиентскую часть.

Клиентская часть реализует интерфейс приложения, через который пользователь может делать запросы к серверу / базе данных.

Клиентская часть имеет следующие страницы:

- Регистрация;
- Авторизация;
- Главная;
- Создание протокола;
- Экспертная оценка;

- Таблица результатов;

Регистрация – Чтобы обучающийся смог пользоваться приложением, ему необходимо зарегистрироваться. Для этого на странице регистрации ему необходимо заполнить четыре обязательных поля: имя, фамилия, почта, пароль.

Авторизация – Страница с двумя обязательными полями (логин и пароль), посредством которой пользователь авторизуется в приложении.

Главная – навигационная страница, отображающая список всех протоколов, созданных пользователем, кнопку для создания нового протокола и поле ввода номера протокола (для подключения к протоколу в роли эксперта).

Создание макета протокола – страница имеет поле для ввода ФИО авторов, которых будут оценивать эксперты и форму для создания критериев оценки. Форма для создания критериев оценки имеет два поля ввода: название критерия и вес критерия.

Экспертная оценка – Страница имеет: выпадающий список, для выбора автора, оценка которого будет производиться; Формы с критериями оценки. Формы имеют: название критерия и четыре кнопки для оценки, с цифрами от нуля до трех.

Таблица результатов – страница доступная только создателю протокола. Статистика имеет выпадающий список с именами авторов и таблицу результатов. При выборе автора из выпадающего списка ниже отображается таблица с оценками экспертов и средним баллом за оцениваемую работу.

3.6. Форматы входных и выходных данных.

На вход пользователь передает текстовые данные: логин, пароль, ФИО. При создании формы протокола для экспертной оценки пользователь также передает текстовые данные: название критерия, раздел, вес критерия.

На выходе пользователь (создатель протокола) получает таблицу с текстовыми данными из базы и xml файл с таблицей доступный для скачивания.

3.7. Источники данных и порядок их ввода в систему (программу), порядок вывода, хранения.

Источником данных является не реляционная база данных MongoDB.

3.8. Порядок взаимодействия с другими системами, возможности обмена информацией.

Не предусмотрено.

3.9. Меры защиты информации.

Приложение должно шифровать пароли пользователей при помощи библиотеки bcrypt.js.

4. Требования к выполняемым системой функциям.

4.1. Общая характеристика пользовательского интерфейса.

- Система должна корректно отображать пользовательский интерфейс при разрешении экрана от 1024x768 пикселей и больше для десктопных устройств, при разрешении меньше 1024x768 интерфейс должен перестроиться под мобильную версию приложения;
- Интерфейс пользователя должен быть представлен на русском языке;

4.2. Размещение информации на странице, дизайн сайта.

Diagram of a login page. In the top-left corner, there is a purple rectangular box labeled "LOGO". Centered on the page is a white rectangular box with a black border. Inside this box, the title "Войдите для экспертной оценки" is centered at the top. Below the title, there are two input fields: the first is labeled "Email" and the second is labeled "Пароль". At the bottom of the box, there are two buttons: a purple button labeled "Войти" and a blue button labeled "Регистрация".

Рис. 7. Страница авторизации

Diagram of a registration page. In the top-left corner, there is a purple rectangular box labeled "LOGO". Centered on the page is a white rectangular box with a black border. Inside this box, the title "Создание аккаунта" is centered at the top. Below the title, there are four input fields: the first is labeled "Имя", the second is labeled "Фамилия", the third is labeled "Email", and the fourth is labeled "Пароль". Below the "Пароль" field, there is a blue hyperlink labeled "У меня уже есть аккаунт". At the bottom of the box, there is a blue button labeled "Зарегистрироваться".

Рис. 8. Страница авторизации

LOGO

Эксперт: имя фамилия

Выйти

Подключиться к протоколу для оценки

Код протокола ...

Подключиться

Мои протоколы

Номер протокола	Дата создания	Статистика	Удалить
csavc6b	15.02.2020		
x7jv4ab	16.02.2020		
f05ffac	21.03.2020		

Создать протокол +

Рис. 9. Главная страница

LOGO

Эксперт: имя фамилия

Выйти

Выберите автора для оценки

Выберите автора ...

Критерии:

Титул критерия оценки
☐ 0 ☐ 1 ☐ 2 ☐ 3

Титул второго критерия оценки
☐ 0 ☐ 1 ☐ 2 ☐ 3

Титул третьего критерия оценки
☐ 0 ☐ 1 ☐ 2 ☐ 3

Отправить

Рис. 10. Страница экспертной оценки

LOGO

Эксперт: имя фамилия

Выйти

Авторы:

Введите имя...

Критерии оценки:

Наименование критерия:

Вес критерия:

X

Наименование критерия:

Вес критерия:

+

Создать

Рис. 11. Страница создания протокола

LOGO

Эксперт: имя фамилия

Выйти

Список авторов:

Выберите автора...

Сохранить в XLSX

Критерии	Вес	Первый автор	Второй автор	Третий автор	Средняя оценка
Первый критерий	3	2	3	2	2,33
Второй критерий	5	1	2	2	1,66
Третий критерий	3	3	3	3	3
Четвертый критерий	4	3	3	2	2,66
Общая оценка					3,88

Рис. 12. Страница итоговой статистики по авторам

4.3. Особенности ввода информации пользователя, представление входных данных.

4.3.1. Требования к странице регистрации.

- Предупреждать пользователя, если имя или фамилия не введены;
- Проверять корректность введенной почты;
- Проверять почту на уникальность;
- Проверять, пароль на минимальную длину в 6 символов;
- Выводить сообщение пользователю об успешном или неуспешном прохождении валидации.

4.3.2. Требования к странице авторизации.

- Проверять корректность введенной почты;
- Проверять: если введенная комбинация логина и пароля верна (занесена в базу, посредством регистрации) - пользователь получает доступ к приложению.

4.3.3. Требования к главной странице.

- Показывать пользователю навигационную шапку страницы с логотипом и кнопкой «выход»;
- При нажатии на логотип перенаправлять на главную страницу.
- Показывать пользователю список всех созданных им ранее протоколов;
- Показывать пользователю поле ввода номера протокола, для подключения к чьему-либо протоколу оценки;
- Проверять введенный номер протокола. Если под таким номером ничего не найдено – сообщить пользователю, в ином случае подключиться к странице протокола в роли эксперта;
- Показывать пользователю кнопку «+ Создать протокол», которая перенаправляет на страницу создания протокола.

4.3.4. Требования к странице создания протокола.

- При заполнении списка авторов, автора можно добавить, как при клике на кнопку «добавить», так и по нажатию кнопки Enter на клавиатуре;
- В форме создания критерия поле «название» не может быть пустым;
- В форме создания критерия поле «вес» содержит число от 0 до 5, по умолчанию 0. Поле не может быть пустым;

- При нажатии Кнопки «+» добавляется еще одна форма критерия;
- При нажатии кнопки «Создать протокол» пользователя перенаправляет на главную страницу, новый протокол появляется первым в списке ранее созданных протоколов.

4.3.5. Требование к странице экспертной оценки.

- Эксперт может выбрать только одного автора проекта, для оценки в один момент времени.
- Если эксперт уже отправил свою оценку по конкретному автору, то он не может отправить ее повторно.
- Пользователь может нажать только на одну из трех кнопок, с цифрами от нуля до трех, для оценки.
- Пока пользователь не нажал кнопку «отправить», он может изменить свою оценку по критерию.

4.3.6. Требование к странице статистики.

- Должна иметь выпадающий список авторов. При выборе автора из выпадающего списка, ниже отображается таблица с экспертными оценками его работы.
- В шапке таблицы должна быть указана фамилия автора
- Должны быть реализованы следующие столбы: критерии оценки, веса критериев, эксперты, средняя оценка от эксперта.
- Внизу таблицы должна быть написана вычисленная общая оценка за работу.
- Страница статистики должна иметь кнопку «скачать в формате xls», при нажатии на которую скачивается файл формата excel, содержащий заполненные протоколы.

5. Требования к документированию.

5.1. Перечень сопроводительной документации.

Руководство пользователя.

5.2. Требования к содержанию отдельных документов.

Не предусмотрено.

6. Порядок сдачи-приемки продукта.

В соответствии с планом выполнения ВКР.

ГЛАВА 2. Реализация мобильной системы пирингового оценивания

2.1 Описание алгоритмов и программной реализации

В соответствии с выбранным стеком технологий в первую очередь была реализована серверная часть приложения. Для этого потребовалось описать файл запуска `app.js`. Первая задача `app.js`: подключить и связать внутри себя фреймворк Express, библиотеку для управления базой данных Mongoose и другие вспомогательные библиотеки и модули. Вторая задача: запуск сервера с помощью Express и описание запросов, по которым представляется возможным удалять, изменять, отправлять и получать данные из базы. Можно прописать все запросы к базе данных внутри файла `app.js`, но для читаемости и будущей масштабируемости кода, они были разбиты на отдельные файлы маршрутов, и подключены в качестве модулей к `app.js`.

Код файла `app.js` выглядит следующим образом:

```
const config = require('config');
const express = require('express');
const mongoose = require('mongoose');

const app = express();
const PORT = config.get('port') || 5000;

app.use('/api/auth', require('./routes/auth.routes'));
app.use('/api/poll', require('./routes/poll.routes'));

async function start() {
  try {
    await mongoose.connect(config.get('mongoUri'), {
      useNewUrlParser: true,
      useUnifiedTopology: true,
      createIndex: true
    });
    app.listen(PORT, () => console.log(`localhost:${PORT}`));
  } catch (err) {
    console.log('Server Error', err.message);
    process.exit(1);
  }
}

start();
```

В первую очередь подключаются: фреймворк Express для создания сервера, библиотека mongoose для организации работы с базой данных и модуль config организующий глобальные конфигурации для приложения. Строчка “const app = Express();” обозначает, что при помощи константы app теперь можно обращаться к методам фреймворка Express. Благодаря этому, используя метод use, подключаем модули маршрутизации. Далее описывается и запускается асинхронная функция "start" со следующей логикой: в блоке try мы ожидаем ответа от сервера с базой данных, передавая ему необходимые для авторизации данные содержащиеся в конфигурации; Если связь установлена – запускаем сервер и прослушиваем порт указанный в конфигурации; Если от сервера базы данных не пришел ответ или сервер по другим причинам не позволяет установить связь – выводим ошибку и завершаем процесс.

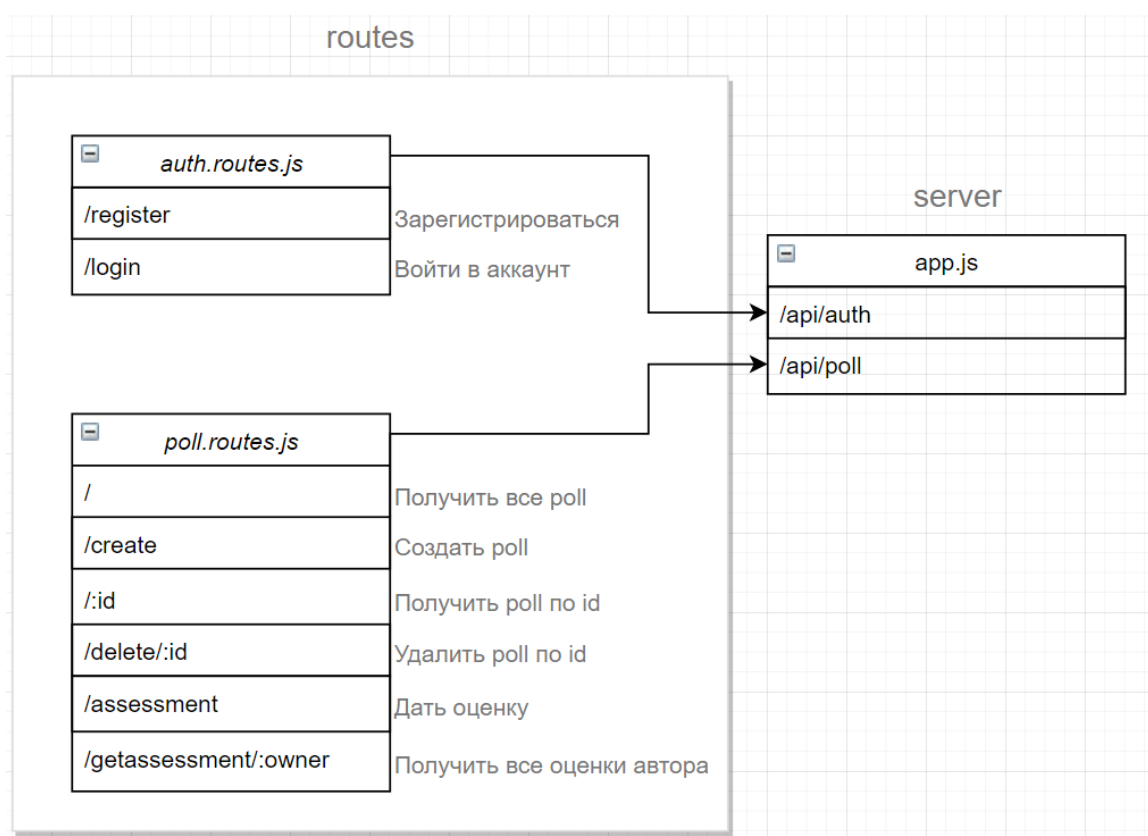


Рис. 13. Схема отношения маршрутов и сервера

Первый файл маршрутов это auth.routes.js. В нем описаны запросы для регистрации и авторизации пользователя.

Второй файл маршрутов это poll.routes.js. Он описывает множество запросов для работы с протоколами оценки:

- Запрос на получение всех созданных пользователем протоколов оценки;
- Запрос на создание протокола оценки;
- Запрос на получение одного протокола по его идентификатору;
- Запрос на удаление протокола по идентификатору;
- Запрос на добавление оценки;
- Запрос на получение всех оценок по выбранному автору работы.

Оба этих файла опираются на модели коллекций (таблиц) и формируют запросы к удаленной базе данных.

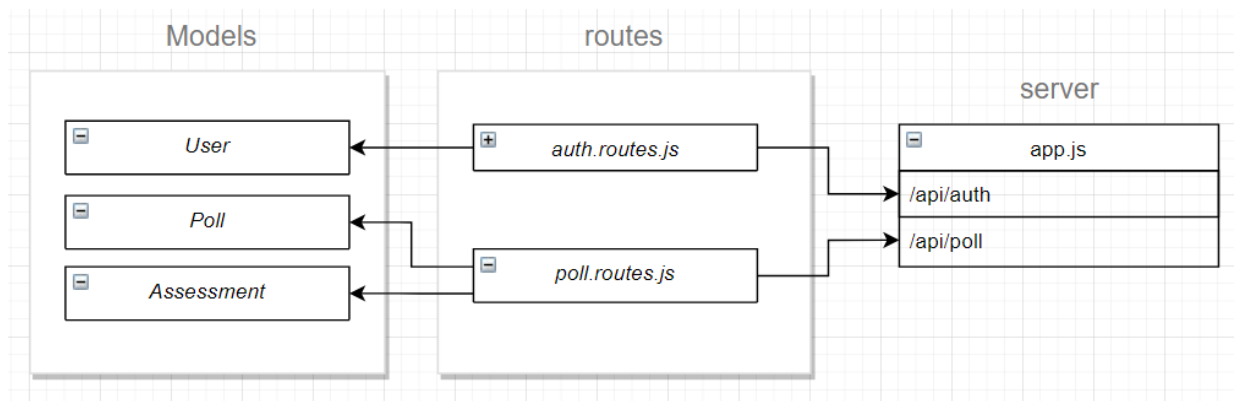


Рис. 14. Схема структуры и отношений внутри серверной части приложения

Для приложения необходимы три сущности: модель пользователя, модель протокола и модель оценки. Связи и схемы данных каждой коллекции видны на диаграмме (Рис. 15).

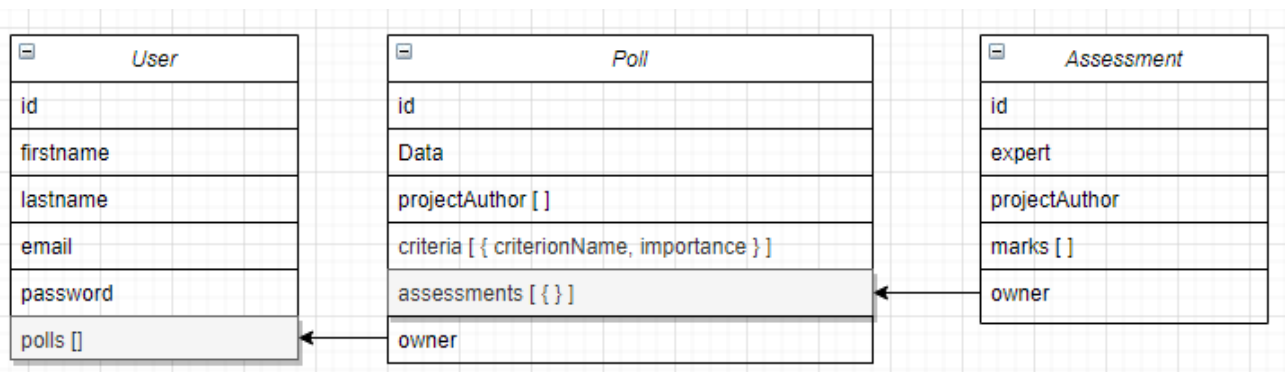


Рис. 15. Схема структуры коллекций и их связей в базе данных.

Каждая модель описывает схему данных и имеет идентификатор для создания связей между коллекциями.

Модель пользователя описывает следующую схему данных:

- Идентификатор пользователя;
- Имя;
- Фамилия;
- Почта;
- Пароль;
- Список протоколов.

Имя, фамилия, почта и пароль задаются пользователем при регистрации, а уникальный идентификатор создается автоматически при успешном ее прохождении. Массив протоколов содержит в себе ссылки на созданные пользователем протоколы и в отличие от предыдущих полей может изменяться.

Схема данных модели протокола описывает следующие поля:

- Идентификатор протокола;
- Дата создания протокола;
- Массив с именами авторов работ;
- Массив с критериями оценки;
- Ссылка на оценки полученные по критериям;
- Ссылка на создателя протокола.

Массив с именами авторов и массив с критериями оценок заполняет пользователь. После создания протокола идентификатор и дата добавляются автоматически, а ссылка на владельца протокола привязывается к создавшему его пользователю. Массив с оценками заполняется динамически неограниченным количеством пользователей, которые присоединились к протоколу и дали свою оценку для выбранных авторов.

Модель оценки состоит из:

- Идентификатор оценки;

- Имя и фамилия эксперта дающего оценку;
- Выбранный для оценки экспертом автор проекта;
- Массив оценок;
- Ссылка на протокол по которому были даны оценки.

Выбор автора проекта и массив оценок заполняет и отправляет пользователь. При отправке в поле expert записывается имя и фамилия эксперта производившего оценку, а идентификатор и ссылка на протокол, как и в других моделях генерируются и добавляются автоматически.

Таким образом, сервер представляет собой API, которое будет принимать HTTP запросы и формировать на их основе запросы к базе данных.

Клиентская часть написана при помощи библиотеки React и реализует одностраничное приложение, то есть перерисовывает только отдельные компоненты, используя единственный HTML-документ как оболочку для всех остальных страниц. Структура клиентской части имеет следующий вид:

Главная страница App.js – это основной файл, который отдается пользователю и внутри которого происходит перерисовка компонентов. App.js импортирует модуль маршрутизации и осуществляет следующую логику: проверяет авторизован ли пользователь, если нет, то ему будет показана страница авторизации и регистрации, в ином случае пользователь попадет на главную страницу и в дальнейшем может переходить на все страницы описанные в файле маршрутизации routes.js.

Директория pages содержит в себе все страницы приложения:

- Главная страница;
- Страница создания протокола;
- Страница протокола для оценки;
- Страница итоговой таблицы оценок;
- Страница авторизации;

- Страница регистрации.

Страницы состоят из множества независимых компонентов, которые можно переиспользовать. Компоненты, в свою очередь, могут использовать внутри себя другие компоненты или разделяться на еще более мелкие.

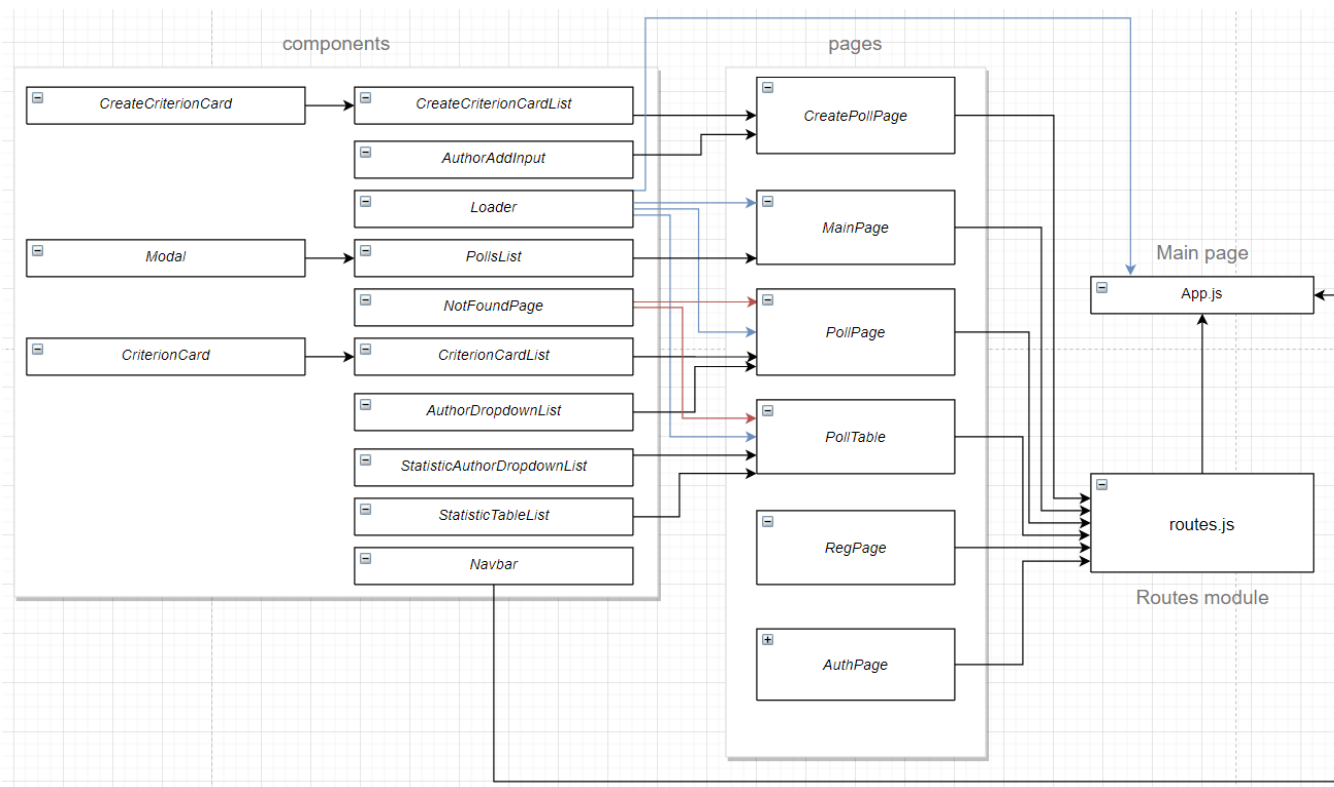


Рис. 16. Схема структуры клиентской части приложения

2.2 Инструкция по использованию системы

Клиентская часть, которая может исполняться как на стационарных компьютерах, так и на мобильных устройствах, имеет следующие страницы:

- Регистрация;
- Авторизация;
- Главная;
- Создание протокола;
- Страница оценки;

- Таблица результатов.

При первом обращении к приложению преподаватель и студент должны однократно зарегистрироваться. При дальнейших обращениях достаточно только авторизоваться.

После регистрации пользователь попадает на главную страницу (рис. 17).

Перед использованием на занятии пирингового оценивая преподаватель должен создать в системе протокол оценивания. При этом он должен нажать экранную кнопку [СОЗДАТЬ ПРОТОКОЛ +] – откроется окно, изображенное на (рис. 18). В нем преподаватель может указать название одного из оценочных критериев и его весовой множитель (по 5-ти-балльной шкале). Посредством кнопки [+] можно добавить следующий критерий, их количество ничем не ограничивается. В поле «Авторы» добавляются (или вставляются копированием) все авторы проектов, которые будут оцениваться. Далее нажимается кнопка [СОЗДАТЬ], после чего система генерирует протокол оценивания с предустановленной 3-х-балльной шкалой оценок (0 – элемент отсутствует; 1 – слабое представление элемента; 2 – элемент представлен хорошо; 3 – очень хорошее представление элемента). Одновременно генерируется код протокола и QR-код для доступа к нему – они отражаются в списке на главной странице.

№	Код протокола	QR	Результаты	Дата создания	удалить
1	oYMPtvUoK			13.05.2020	
2	lYRy4ahE1			10.05.2020	

Рис. 17. Главная страница сервиса «Peer assessment online»

Рис. 18. Страница создания протокола

Список включает:

- код протокола, который можно дать для подключения обучающимся;
- сгенерированный QR код, по которому можно перейти на страницу оценки в протоколе;
- кнопка перехода к таблице результатов;
- текстовое поле с датой создания;
- кнопка удаления протокола.

После перехода студента-эксперта по символьному или QR коду на страницу протокола ему будет предложен список авторов проектов (рис. 4). Задача учащегося выбрать автора и произвести оценку на основе предоставленных критериев и после нажать кнопку отправить.

The screenshot shows a mobile application interface for peer assessment. At the top, the status bar displays the time 11:53, signal strength, and battery level at 53%. The browser address bar shows the URL peer-assessment-online.ru/. The app header features a blue bar with a menu icon on the left and the 'PEER ASSESSMENT' logo on the right. The main content area is titled 'Выберите автора для оценки' (Select an author for evaluation). Below this is a dropdown menu labeled 'Выберите автора'. The section 'Критерии:' (Criteria:) follows, containing three evaluation criteria, each with a rating scale from 0 to 3:

- Вводная часть** (Introductory part): Rating scale 0, 1, 2, 3.
- Описанная диаграмма. Структурность.** (Described diagram. Structurality): Rating scale 0, 1, 2, 3.
- Заключение** (Conclusion): Rating scale 0, 1, 2, 3.

At the bottom of the form is a green button labeled 'ОТПРАВИТЬ >' (SEND >).

Рис. 19. Страница оценивания

Просмотр итогов оценивания возможен для преподавателя при нажатии пиктограммы «Результаты». Поскольку результаты относятся к конкретному автору проекта, для формирования таблицы результатов необходимо указать его из заведенного ранее списка. В Таблице высчитываются средние оценки от экспертов по каждому критерию и общая оценка с учетом весов (рис. 5). Таблица допускает выгрузку и сохранение в формате MSExcel.

PEER
ASSESSMENT

Эксперт: Имя Фамилия_7

Выйти

Список авторов:

Имя автора_1

СОХРАНИТЬ В XLSX

Критерии	Вес	Эксперт_1 Имя	Эксперт_2 Имя	Эксперт_3 Имя	Эксперт_4 Имя	Эксперт_5 Имя	Средняя
Вводная часть	3	2	1	3	2	3	2.2
Описанная диаграмма. Структурность.	5	3	2	3	2	3	2.6
Заключение	4	3	3	2	3	3	2.8
Общая оценка							4.28

Рис. 20. Страница таблицы результатов

Присоединяться к сервису преподаватели и учащиеся могут с персональных компьютеров или мобильных устройств с установленным браузером и доступом в интернет.

Следует отметить, что студент имеет возможность выступить в качестве преподавателя — составить свой протокол оценивания и использовать его, например, в работе с учащимися на педагогической практике или при выполнении своего исследования. Этот момент представляется для использования сервиса в педагогическом вузе.

2.3 Результаты апробации

Экспертиза проводилась при помощи разработанного приложения. Специалистам был выдан код протокола с критериями направленными на оценку разработанного продукта.

Общее оформление приложения (эстетика)	<input type="radio"/> 0	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3
Понятность пользовательского интерфейса	<input type="radio"/> 0	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3
Быстродействие приложения	<input type="radio"/> 0	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3
Удобство работы с приложением с мобильного устройства	<input type="radio"/> 0	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3
Полезность приложения в учебной работе	<input type="radio"/> 0	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3
Могли бы Вы использовать приложение в своих учебных курсах?	<input type="radio"/> 0	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3
Общее впечатление	<input type="radio"/> 0	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3

Рис. 21. Критерии оценки, предложенные экспертам при апробации разработанного приложения

По отправленным оценкам сформировалась итоговая таблица с общей оценкой специалистов.

Критерии	Вес	Star Boris	Сардак Любовь	Арбузов Сергей	Слепухин Александр	Средняя
Общее оформление приложения (эстетика)	5	3	2	2	3	2.5
Понятность пользовательского интерфейса	5	3	3	2	3	2.75
Быстродействие приложения	5	3	3	3	3	3
Удобство работы с приложением с мобильного устройства	5	3	3	3	3	3
Полезность приложения в учебной работе	5	3	3	1	3	2.5
Могли бы Вы использовать приложение в своих учебных курсах?	5	3	3	1	2	2.25
Общее впечатление	0	3	3	2	3	2.75
Общая оценка						4.44

Рис. 22. *Итоговая таблица с оценками экспертов*

Также в письменном виде были получены следующие замечания и пожелания:

- Не хватает функции редактирования созданного протокола для правки критериев. Также необходимо учесть, что после изменения веса, в итоговой таблице пересчитывались результаты с уже полученными оценками.
- Добавлять большой список авторов на данный момент не представляется возможным. Хотелось бы иметь функционал позволяющий инициализировать список, к примеру: посредством импортирования текстового файла с именами авторов.

- При нажатии на кнопку добавления критерия оценки, мгновенно появляется новый: его выведение можно сделать плавным, анимированным.

По итогу апробации разработанного веб-приложения, получены высокие оценки специалистов. Общая оценка, полученная в сформированной итоговой таблице, равна 4,44 по пятибалльной шкале. Таким образом, после незначительных доработок, приложение рекомендовано к использованию в учебном процессе.

Заключение

Результатом разработки стало веб-приложение для мобильной пиринговой оценки. Приложение позволяет проводить пиринговую оценку в любом месте, где обучающиеся могут подключиться к сети интернет со своих мобильных устройств.

В процессе выполнения работы в рамках сформулированных задач было сделано:

1. На основании проведенного анализа различных информационных источников, были выявлены и проанализированы существующие подходы и средства для реализации метода пиринговой оценки посредством ИКТ.
2. В результате анализа существующих технологий и средств реализации пиринговой оценки была обоснована актуальность и необходимость разработки приложения.
3. Разработанная система мобильной пиринговой оценки соответствует требованиям технического задания
4. Подготовлена техническая и сопроводительная документация с примерами использования разработанного приложения для решения практических задач.

Таким образом, следует считать, что результаты разработки соответствуют всем требованиям технического задания, поставленная цель достигнута. Работа носит законченный характер.

Список информационных источников

1. Акт министерств и ведомств "Об утверждении профессионального стандарта «Педагог (педагогическая деятельность в сфере дошкольного, начального общего, основного общего, среднего общего образования) (воспитатель, учитель)" от 18 октября 2013 г. №544н // Российская газета.
2. Архитектура React: несколько шаблонов и практик, упрощающих работу с приложением // webformymself.com URL: <https://webformymself.com/5-luchshix-praktik-v-arxitecture-react/> (дата обращения: 30.04.2020).
3. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению; с Изм. N 1, утвержденным в июне 1981 г.; Введ. 1980-01-01.
4. ГОСТ 19.504-79. Руководство программиста. Введ. 1980-01-01. М.: Стандартинформ. 2 с. (Межгос. стандарт.. Единая система программной документации. Требования к содержанию и оформлению).
5. ГОСТ 19.505-79. Руководство оператора. Введ. 1980-01-01. М.: Стандартинформ. 2 с. (Межгос. стандарт.. Единая система программной документации. Требования к содержанию и оформлению).
6. ГОСТ 34.602-89 Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы. Взамен ГОСТ 24.201-85 Введ. 1990-01-01.
7. ГОСТ 7.1-2003. Библиографическая запись. Библиографическое описание. Взамен ГОСТ 7.0-84; Введ. 2004-07-01. Межгос. совет по стандартизации, метрологии и сертификации. М.: Изд-во стандартов. 111 с. (Межгос. стандарт. Система стандартов по информации, библиотечному и издательскому делу. Общие требования и правила составления).
8. ГОСТ Р 52653-2006. Термины и определения. Введ. 2008-07-01. М.: Стандартинформ. 12 с. (Национальный стандарт Российской Федерации. Информационно-коммуникационные технологии в образовании).
9. ГОСТ Р 53620-2009. Электронные образовательные ресурсы. Введ. 2009-12-15. М.: Стандартинформ. 12 с. (Национальный стандарт Российской

- Федерации. Информационно-коммуникационные технологии в образовании. Общие положения).
- 10.ГОСТ Р 7.0.83-2013. Электронные издания. Основные виды и выходные сведения. С изм. и допол. в ред. от 12.09.2018 г.; Введ. 2013-10-15. М.: Стандартинформ. 16 с. (Система стандартов по информации, библиотечному и издательскому делу).
 - 11.Грокаем алгоритмы. Иллюстрированное пособие для программистов и любопытствующих Бхаргава Адитья. - Санкт-Петербург: Питер, 2019.
 - 12.Жидкова Р.А – Современные методы оценивания результатов обучения // Известия ПГПУ им В.Г Белинского. 2012 №28. С. 779-782
 - 13.Кайл Бэнкер MongoDB в действии. - Москва: ДМК-Пресс, 2012.
 - 14.Клиент - сервер - Википедия // ru.wikipedia.org URL: https://ru.wikipedia.org/wiki/Клиент_—_сервер (дата обращения: 10.04.2020).
 - 15.Маршрутизация в Express // expressjs.com URL: <http://expressjs.com/ru/guide/routing.html> (дата обращения: 25.04.2020).
 - 16.Мешнина Е.Ю., Использование взаимооценки и взаимного обучения в системе профессиональной подготовки преподавателей иностранных языков // Тезисы конференции «70-е Дни Науки студентов НИТУ “МИСиС”» // URL: <http://xpir.fcntp.ru/theses/Ispolzovanie-metodov-vzaimoocenki-i-vzaimnogo-obucheniya-v-sisteme-professionalnoi-podgotovki-prepodavatelei-inostrannih-yazikov> (дата обращения: 02.05.2020).
 - 17.Паттерны проектирования в JavaScript / Блог компании RUVDS.com / Хабр // habr.com URL: <https://habr.com/ru/company/ruvds/blog/427293/> (дата обращения: 04.05.2020).
 - 18.Персональные учебные среды в дистанционном обучении / В.Б. Артеменко, А.Г. Карпа, О.И. Полотай // Управляющие системы и машины. 2012. № 2. С. 20-26.
 - 19.Приказ Минобрнауки России "Об утверждении федерального государственного образовательного стандарта высшего образования -

- магистратура по направлению подготовки 44.04.01 Педагогическое образование" от 22.02.2018 №126 // Российская газета.
20. Применение пириговой оценки письменных работ в потоковых очных курсах - Национальный исследовательский университет «Высшая школа экономики». URL: <https://publications.hse.ru/mirror/pubs/share/folder/o9015s1v7w/direct/196548998> (дата обращения: 03.04.2020).
21. Семенова И. Н., Слепухин А. В. Методика использования информационно-коммуникационных технологий в учебном процессе. Ч. 2. Методология использования информационных образовательных технологий : учеб. пособие / Урал. гос. пед. ун-т. Екатеринбург, 2013. 144 с.
22. Система управления обучением // ru.wikipedia.org URL: https://ru.wikipedia.org/wiki/Система_управления_обучением (дата обращения: 10.04.2020).
23. Современный учебник JavaScript // learn.javascript.ru URL: <https://learn.javascript.ru/> (дата обращения: 04.04.2020).
24. Стоян Стефанов JavaScript. Шаблоны. - Санкт-Петербург: Символ, 2017.
25. Таксономия педагогических целей Б.Блума — ТолВИКИ // wiki.tgl.net.ru URL: http://wiki.tgl.net.ru/index.php/Таксономия_педагогических_целей_Б.Блума (дата обращения: 21.04.2020).
26. Тидвелл Дженифер Разработка пользовательских интерфейсов. - Санкт-Петербург: Питер, 2011.
27. Шелли Пауэрс Изучаем Node. Переходим на сторону сервера. - Санкт-Петербург: Питер, 2014.
28. about Materialize // materializecss.com URL: <https://materializecss.com> (дата обращения: 29.04.2020).
29. bcryptjs - npm // npmjs.com URL: <https://www.npmjs.com/package/bcryptjs> (дата обращения: 01.05.2020).

- 30.Chrome DevTools // developers.google.com URL:
<https://developers.google.com/web/tools/chrome-devtools> (дата обращения:
10.05.2020).
- 31.Mongoose v5.9.16: Schemas // mongoosejs.com URL:
<https://mongoosejs.com/docs/guide.html> (дата обращения: 26.04.2020).
- 32.Node.js - Википедия // ru.wikipedia.org URL:
<https://ru.wikipedia.org/wiki/Node.js> (дата обращения: 09.04.2020).
- 33.React - JavaScript библиотека для создания пользовательских интерфейсов //
ru.reactjs.org URL: <https://ru.reactjs.org/> (дата обращения: 20.04.2020).
- 34.SheetJS/sheetjs: SheetJS Community Edition -- Spreadsheet Data Toolkit //
github.com URL: <https://github.com/sheetjs/sheetjs> (дата обращения:
12.05.2020).
- 35.shortid - npm // npmjs.com URL: <https://www.npmjs.com/package/shortid> (дата
обращения: 05.05.2020).

ПРИЛОЖЕНИЯ

Приложение 1.

Приложение 2.